

LIBRO DE TEXTO

NIVEL BASICO

MÓDULO III :
ENTRADA/SALIDA
DE POTENCIA ARDUINO Y
TINKERCAD



ROBÓTICA

2022





Contenido

1. Dispositivos de entrada y salida Arduino	1
1.1. Funciones para codificar las entradas y salidas digitales de Arduino	2
2. Clasificación de los dispositivos de entrada y salida Arduino	3
3. Practicas con Arduino	7
3.1. Encienda el led interno del Arduino (opción 1)	7
3.2. Encienda el led interno del Arduino (opción 2)	10
4. Protoboard	12
4.1. Descripción de la placa de pruebas (repaso)	12
4.2. Control de LED con Arduino protoboard	13
4.3. Control de 2 LED con Arduino	15
4.4. Control de 3 LED con Arduino	16
4.5. Control de 1 LED con Arduino opción 2	18
4.6. Control de 2 LED con Arduino opción 2	19
4.7. Control de 3 LED con Arduino opción 2	22

Modulo III Entrada/Salida De Potencia Arduino Y Tinkercad	Tiempo de aprendizaje
	6 horas

OBJETIVOS

- ✓ Conocer los dispositivos de entrada y salida Arduino
- ✓ Codificar las funciones de entrada y salida digital de Arduino
- ✓ Conocer la clasificación de los dispositivos de entrada y salida en Arduino
- ✓ Realizar los ejercicios planteados de manera correcta, elaborando circuitos en Tinkercad y codificación en Arduino

1. Dispositivos de entrada y salida Arduino

Una placa Arduino tiene los pines comunes que son 5 Voltios y GND (0 Voltios), pero a su vez utiliza más pines como entradas y salidas siendo el 5V la representación del 1 lógico, que puede ser utilizado como salida entregando 5V o como entrada recibéndolo. con el 0 lógico, en otras palabras, no entrega voltaje y no recibe voltaje en dicho pin.

Una señal digital es una variación de voltaje entre $-V_{cc}$ a $+V_{cc}$ sin pasar por los valores intermedios. Por lo tanto, una señal digital dispone solo de dos estados. Al valor inferior de tensión $-V_{cc}$ le asociamos un valor lógico LOW o '0', mientras que al valor superior $+V_{cc}$ le asociamos HIGH o '1' lógico.

En realidad, una entrada digital realiza una comparación de la medición con un valor de tensión umbral. Si el valor medido es superior a la tensión umbral se devuelve HIGH, y si es inferior LOW. El valor de la tensión umbral varía de un autómata a otro, e incluso no tiene porque permanecer constante a lo largo del tiempo.



1.1. Funciones para codificar las entradas y salidas digitales de Arduino

Las funciones de programación para tratar las entradas y salidas digitales en Arduino son:

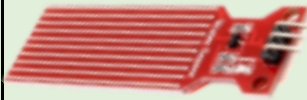
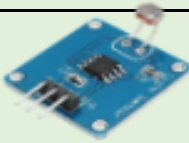
1. pinMode() sirve para configurar en el pin como se va a comportar como una entrada o una salida.
2. digitalWrite() funciona describiendo un valor HIGH o LOW (Alto o bajo) en el pin.
3. digitalRead() funciona leyendo el valor del pin correspondiente como HIGH o LOW

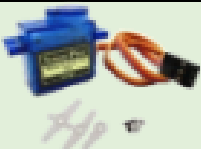
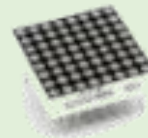
Funciones para tratar las entradas y salidas analógicas de Arduino

1. analogReference() configura el voltaje usado para la entrada analógica.
2. analogRead() lee el valor del pin analógico.
3. se puede escribir un valor analógico (onda PWM) al pin especificado.

2. Clasificación de los dispositivos de entrada y salida Arduino

No.	PIEZA	IMAGEN	TIPO
1	Porta Baterias AA de 6		Accesorio de Potencia
2	Interruptor sencillo I/O		Entrada
3	Sensor a distancia ultrasonico		E
4	Sensor Infrarojo		E
5	Bluetooth sensor		E
6	Joystick Shield		E
7	Pulsadores		E
8	LDR		E
9	Potenciometro con perilla 10k		E
10	Sensor de vibracion, interruptor de inclinacion		E
11	Transistor con sensor de temperatura		E

No.	PIEZA	IMAGEN	TIPO
12	Sensor detector de agua/humedad		E
13	Modulo rele Arduino		E
14	Sensor de movimiento arduino		E
15	Modulo Jostick 4 pines		E
16	Modulo Laser		E
17	Modulo Sensor fotosensible		E
18	Modulo Sensor infrarojo piroelectrico		E
19	Modulo Pulsador Tres pines		E

No.	PIEZA	IMAGEN	TIPO
1	Motor con caja reductora		Salida
2	Servo motor SG90		S
3	LED ROJO		S
4	LED RGB 4 pines		S
5	Pantalla 7 segments 4 digitos		S
6	Pantalla 7 segments 1 digitos		S
7	Matriz LED 8*8		S
8	Motor DC		S

No.	PIEZA	IMAGEN	TIPO
9	Modulo LED trafico light 5 pines		S
10	Modulo Sound sensor		S
11	Modulo Buzzer		S
12	Modulo Pantalla LCD 16*2		S
13	Modulo motor vibrador 3 pines		S
14	Modulo motor 4 pines CI		S
15	Modulo ShiEld neo pixeles 6 pines		S
16	Modulo display 7 segmentos 4 pines		S

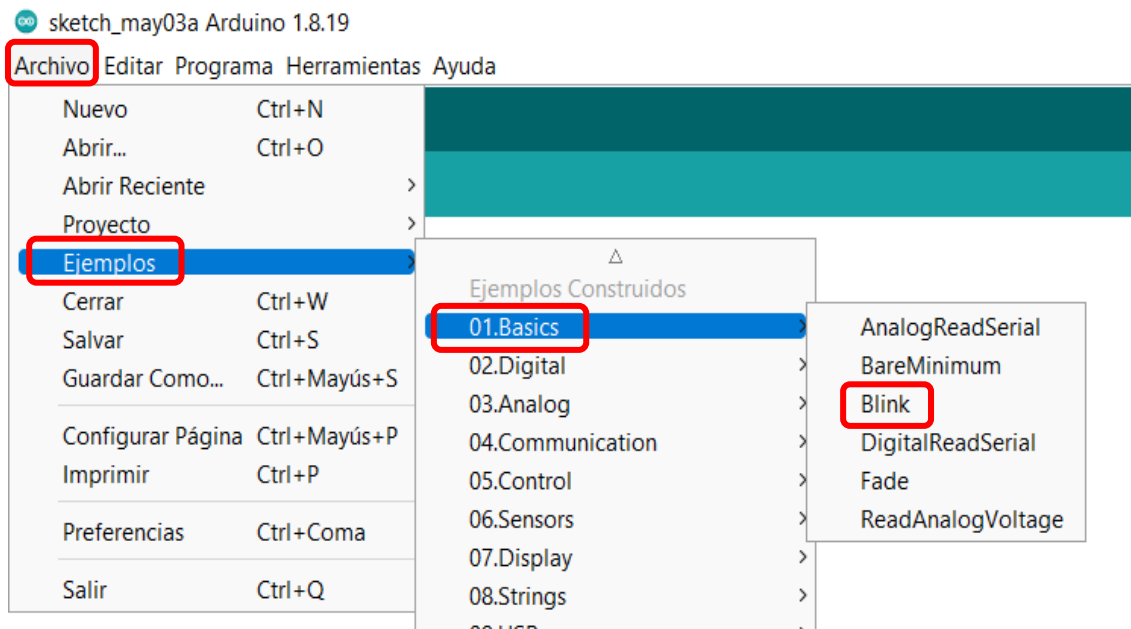
3. Practicas con Arduino

3.1. Encienda el led interno del Arduino (opción 1)

OBJETIVO: Por medio de la placa Arduino UNO y su interfaz cargar la codificación del programa Blink del recurso ejemplos y ejecutarlo.

MATERIALES: Placa Arduino UNO, Computadora, Cable USB, programa Arduino instalado.

- Comenzaremos con el ejercicio del cargado de un programa a la placa Arduino.
- a) Abra la interfaz de Arduino
- b) Una vez abierta, busque en el menú principal y seleccione la opción Archivo, seguidamente la opción ejemplos, 01.Basics y por ultimo Blink como lo muestra la siguiente imagen:



Inmediatamente la codificación correspondiente a ese programa será cargada en la interfaz y lista para subirla a la placa y ejecutarla.



```

Blink Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

Blink $

/*
  Blink

  Enciende el LED interno de la placa durante un segundo, luego apaga durante un segundo de manera repetida.
  La mayoría de los Arduinos tienen un LED a bordo que puede controlar. Sobre la ONU, MEGA y ZERO
  está conectado al pin digital 13, en MKR1000 en el pin 6. LED_BUILTIN está establecido en
  el pin LED correcto independientemente de la placa que se utilice.
  Si desea saber a qué pin está conectado el LED integrado en su Arduino
  modelo, consulte las especificaciones técnicas de su placa en:
  https://www.arduino.cc/en/Main/Products

  modificado el 8 de mayo de 2014
  por Scott Fitzgerald
  modificado el 2 sep 2016
  por Arturo Guadalupi
  modificado el 8 sep 2016
  por Colby Newman

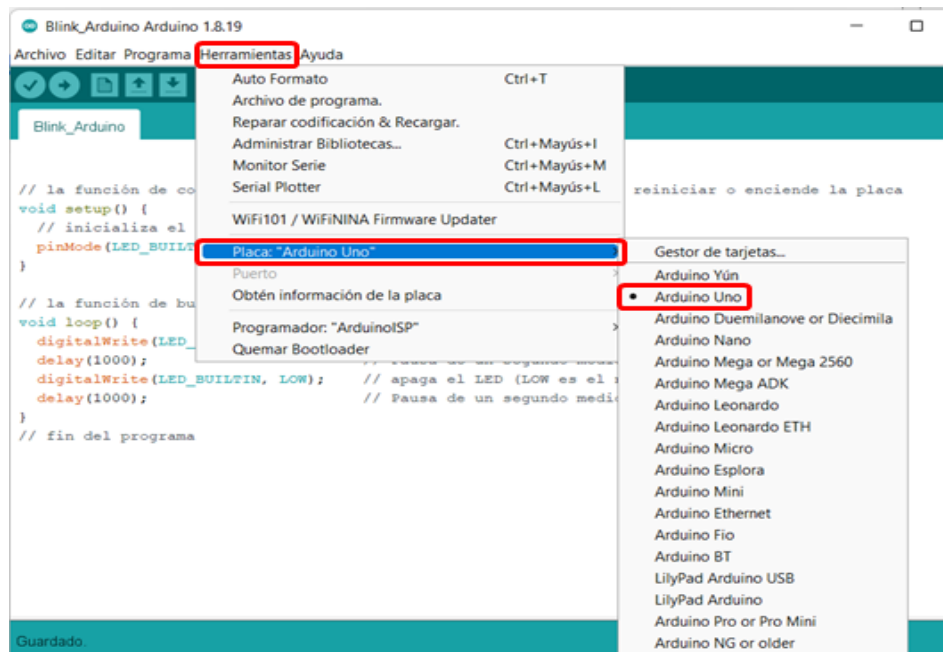
  Este código de ejemplo es de dominio público.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// La función de instalación(void setup() ) se ejecuta una vez al presionar Restablecer o encender la placa
void setup() {
  // inicializa el pin digital LED_BUILTIN como salida.
  pinMode(LED_BUILTIN, OUTPUT);
}

// la función de bucle se ejecuta una y otra vez de manera indefinida
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // enciende el LED (HIGH es el voltaje encendido)
  delay(1000); // hace una pausa de un segundo
  digitalWrite(LED_BUILTIN, LOW); // apaga el LED (LOW es el voltaje apagado)
  delay(1000); // hace una pausa de un segundo
}
  
```

- Conecte el Arduino uno al pc
- Luego en el menú de herramientas dele click a la opción placa Arduino y seleccione la placa a utilizar que será Arduino uno.



- Luego seleccione el menú de herramientas de le a la opción puerto COM: y seleccione la opción donde aparece la opción de Arduino uno

Así su Arduino ya estará comunicado con el programa y podrá subir los códigos.

- Luego cargue el código al Arduino



- Observe como el led interno Arduino enciende intermitentemente de acuerdo a los valores establecidos en el ejemplo
- Modifique los tiempos del led tanto de encendido como de apagado para que entienda el funcionamiento
- Puede cambiar en delay (500)

3.2. Encienda el led interno del Arduino (opción 2)

OBJETIVO: Por medio de la placa Arduino UNO y su interfaz, copiar código en Arduino para la opción dos de encender led interno Arduino y ejecutarlo.

MATERIALES: Placa Arduino UNO, Computadora, Cable USB, programa Arduino instalado.

- a) Abra la interfaz de Arduino
- b) Una vez abierta, copie el código Arduino así como esta

Código de Ejemplo:

```
/*
Programa Inicial
Enciende un LED por un segundo y lo apaga por el mismo tiempo
*/
//Función principal iniciada con la palabra reservada void setup()
void setup()                                // Se ejecuta cada vez que el Arduino
se inicia
{
```

```
pinMode(13, OUTPUT);           // Inicializa el pin 13 de la placa como
una salida
}

//Función cíclica iniciada con la palabra reservada void loop()
void loop()                     // Esta función se mantiene ejecutando
{                               // cuando este energizado el Arduino
    digitalWrite(13,HIGH);      // Enciende el LED
    delay(1000);                // Temporiza un segundo (1s = 1000ms)
    digitalWrite(13,LOW);      // Apaga el LED
    delay(1000);                // Temporiza un segundo (1s = 1000ms)
}
// Fin del programa
```

c) Guarde el código y coloque el nombre

d) Verificar

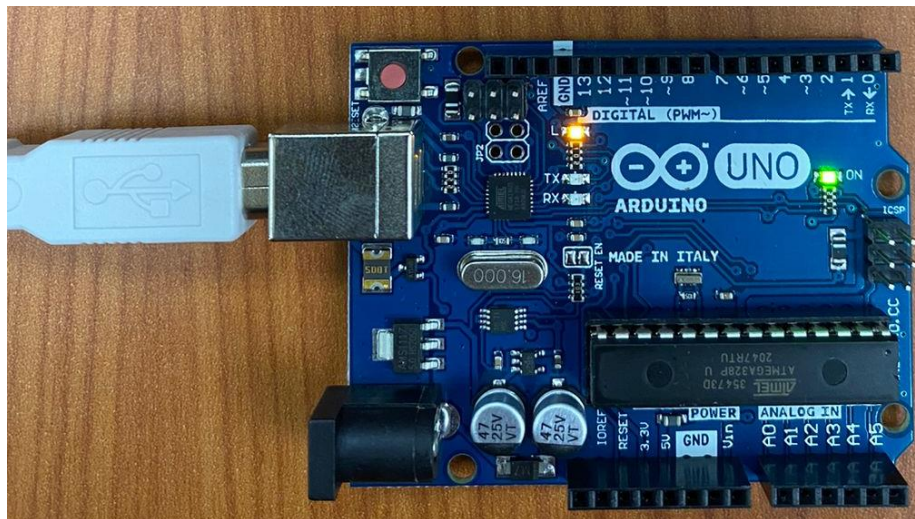


e) CARGA



f) Programa ya en funcionamiento.

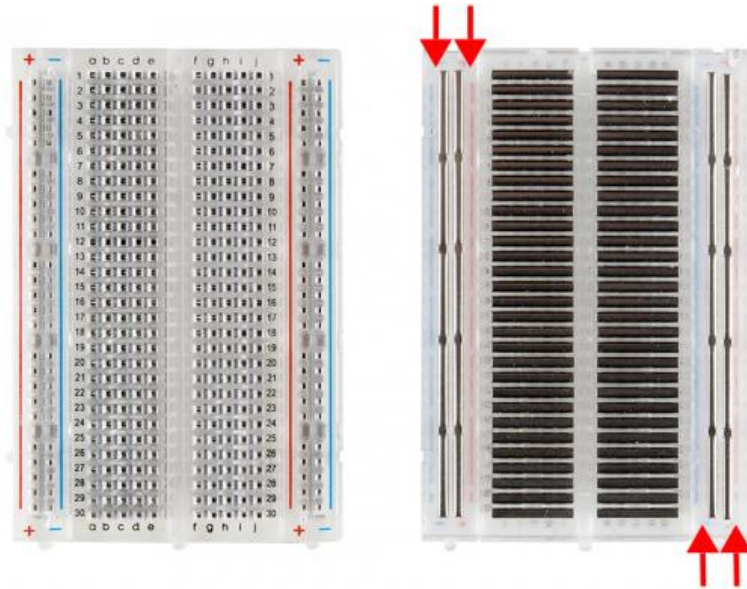
g) Luz amarilla es la que estamos controlando



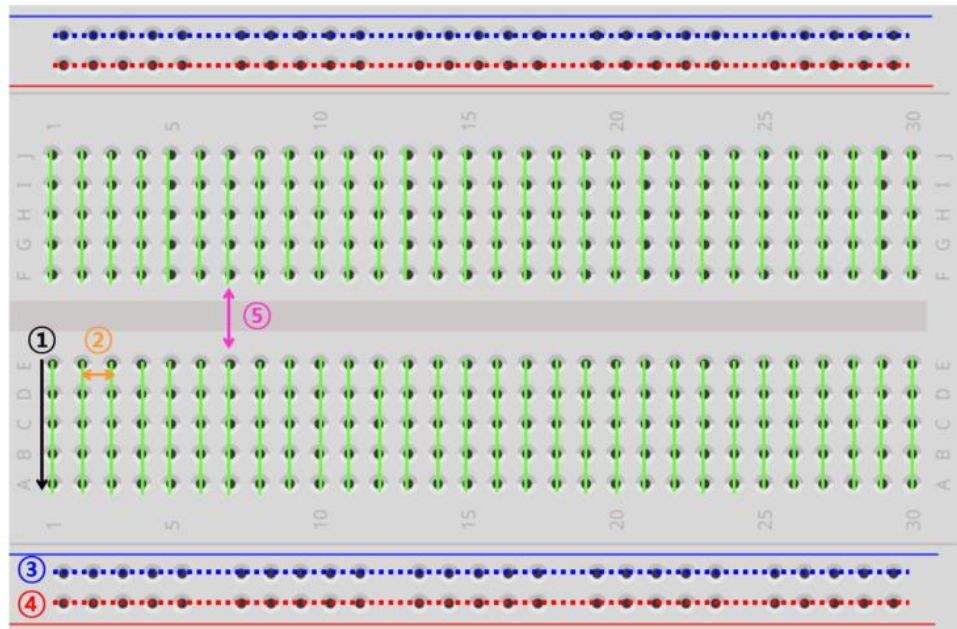
4. Protoboard

4.1. Descripción de la placa de pruebas (repaso)

El protoboard es una placa de pruebas que permite interconectar elementos electrónicos sin la necesidad de soldar componentes. Logrando así que se facilite el armado de circuitos o sistemas electrónicos.



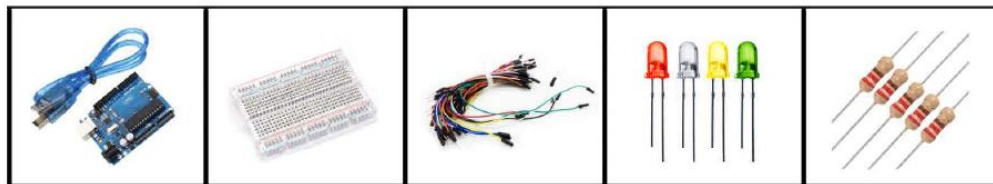
Primero, las líneas verdes de la placa de pruebas tienen cinco agujeros conectados entre sí en la dirección vertical. Si nos fijamos en la flecha negra de (1), cinco agujeros están conectados verticalmente entre sí. Sin embargo, los agujeros horizontales de (2) no están conectados entre sí. La línea azul de (3) y la línea roja de (4) están conectadas a todos los orificios de un eje horizontal largo y recto. Sin embargo, los agujeros en la línea roja y los agujeros en la línea azul no están conectados entre sí. La pieza (5) no está conectada a las partes superior e inferior.



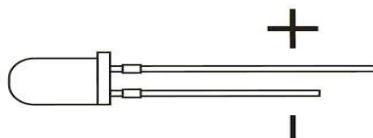
4.2. Control de LED con Arduino protoboard

OBJETIVO: controlar la potencia del LED emitiendo una señal HIGH(1) o LOW(0) a través de un pin digital del Arduino.

MATERIALES:



Placa Uno / Placa de prueba / Cable de puente / LED / 1 Resistencia de 220 ohmios (Ω)



Polaridad del LED: el lado largo es el ánodo (+), el lado corto es el cátodo (-)

CODIFICACIÓN:

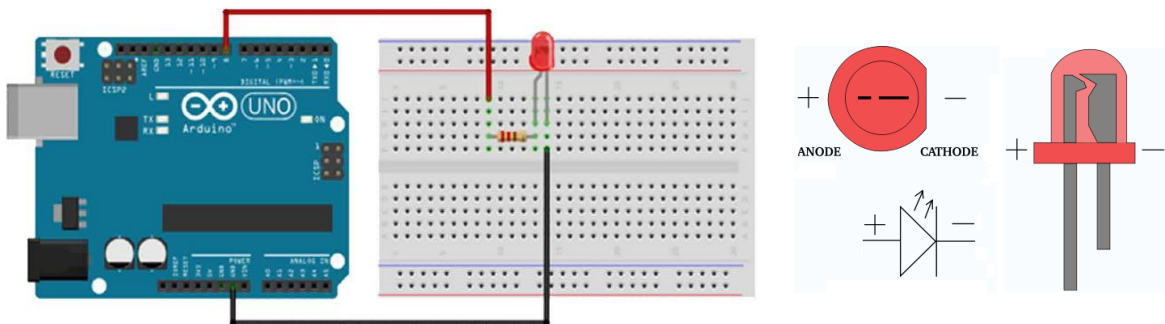
```
void setup() {
  pinMode(8, OUTPUT); // Establece el pin digital 8 como modo
  de salida.
}

void loop() {
  digitalWrite(8, HIGH); // Emita la señal HIGH (5V) al pin 8.
  delay(1000);           // establece un segundo de pausa.
  digitalWrite(8, LOW);  // Salida de la señal LOW (0V) al pin 8.
  delay(1000);           // establece un segundo de pausa
}
```

Después de configurar el pin digital 8 al que está conectado el LED como modo de salida, la alimentación del LED se enciende y apaga repetidamente emitiendo una señal ALTA (5V) durante 1 segundo y una señal LOW (0V) durante 1 segundo.

Una vez que estés seguro que el circuito ha sido cargado con éxito en la placa Arduino uno procedemos con el armado del circuito que está en la siguiente imagen

- Conecte el poste (-) del LED al GND del Arduino.
- Conecte el poste (+) del LED al pin digital 8 del Arduino.
- (Conéctese con una resistencia de 220 ohmios en el medio).
- Una vez terminado guarde, coloque el nombre y verifique si el código está bien.



4.3. Control de 2 LED con Arduino

OBJETIVO: controlar dos LED emitiendo una señal HIGH(1) ó LOW(0) a través de dos pines digital del Arduino.

MATERIALES:



Placa Uno / Placa de prueba / Cable de puente / 2 LED / 2 Resistencias de 220 ohmios (Ω), cables para conexión.

CODIFICACIÓN:

```
void setup() {
  pinMode(8, OUTPUT); // Establece el pin digital 8 como modo
                        // de salida.
  pinMode(9, OUTPUT); // Establece el pin digital 9 como modo
                        // de salida.
}

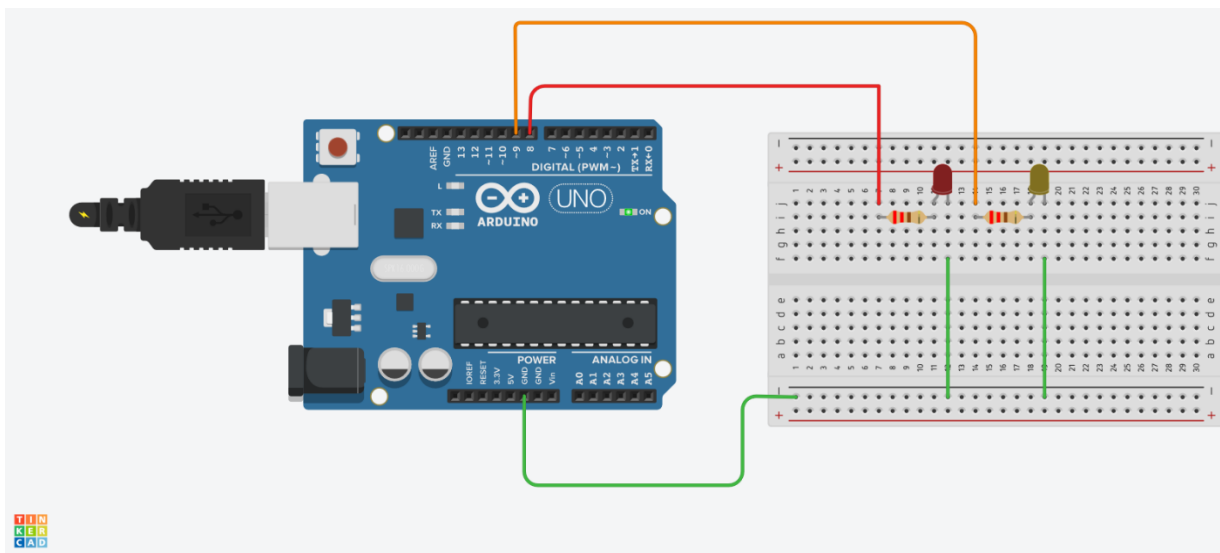
void loop() {
  digitalWrite(8, HIGH); // Emita la señal HIGH (5V) al pin 8.
  delay(1000);           // establece un segundo de pausa.
  digitalWrite(8, LOW);  // Salida de la señal LOW (0V) al pin 8.
  delay(2000);           // establece un segundo de pausa
  digitalWrite(9, HIGH); // Emita la señal HIGH (5V) al pin 9.
  delay(1000);           // establece un segundo de pausa.
  digitalWrite(9, LOW);  // Salida de la señal LOW (0V) al pin 9.
  delay(1000);           // establece un segundo de pausa
}
```

- Una vez terminado guarde, coloque el nombre y verifique si el código está bien.
- Una vez verificado suba la codificación a su Arduino

- Desconecte el Arduino y realice la siguiente conexión

PROCEDIMIENTO:

1. Conectamos dos LED a nuestra Protoboard como se muestra la imagen ánodo con la resistencia y cátodo a tierra.
2. Conectamos dos resistencias de 220 ohm a cada LED.
3. Conectamos la resistencia al pin 8 el LED rojo y al pin 9 al LED amarillo.
4. Verifique q está bien conectado todo espere la autorización del docente y conecte su Arduino a la fuente

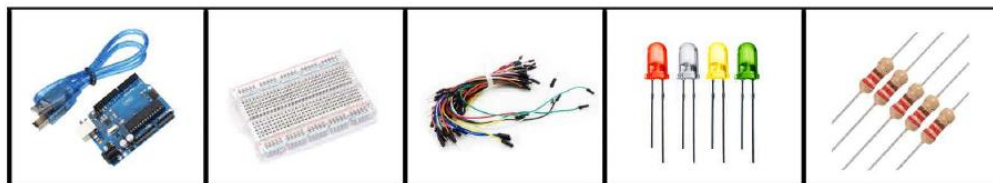


Una vez que estamos seguro que ha sido conectado de forma correcta procedemos con el funcionamiento y análisis del circuito.

4.4. Control de 3 LED con Arduino

OBJETIVO: controlar dos LED emitiendo una señal HIGH(1) o LOW(0) a través de tres pines digital del Arduino.

MATERIALES:





Placa Uno / Placa de prueba / Cable de puente / 3 LED / 3 Resistencias de 220 ohmios (Ω), cables para conexión.

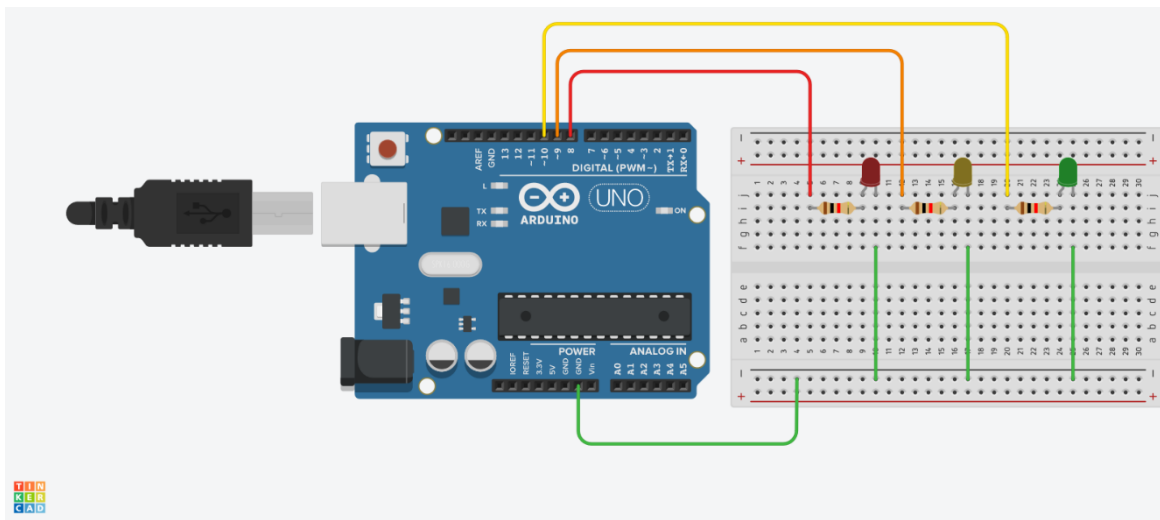
CODIFICACIÓN:

```
void setup() {  
  pinMode(8, OUTPUT); // Establece el pin digital 8 como modo  
  de salida.  
  pinMode(9, OUTPUT); // Establece el pin digital 9 como modo  
  de salida.  
  pinMode(10, OUTPUT); // Establece el pin digital 9 como modo  
  de salida.  
}  
  
void loop() {  
  digitalWrite(8, HIGH); // Emita la señal HIGH (5V) al pin 8.  
  delay(1000);           // establece un segundo de pausa.  
  digitalWrite(8, LOW);  // Salida de la señal LOW (0V) al pin 8.  
  delay(2000);           // establece un segundo de pausa  
  digitalWrite(9, HIGH); // Emita la señal HIGH (5V) al pin 9.  
  delay(1000);           // establece un segundo de pausa.  
  digitalWrite(9, LOW);  // Salida de la señal LOW (0V) al pin 9.  
  delay(1000);           // establece un segundo de pausa  
  digitalWrite(10, HIGH); // Emita la señal HIGH (5V) al pin 10.  
  delay(1000);           // establece un segundo de pausa.  
  digitalWrite(10, LOW); // Salida de la señal LOW (0V) al pin 10.  
  delay(1000);           // establece un segundo de pausa  
}
```

- Una vez terminado guarde, coloque el nombre y verifique si el código esta bien.
- Una vez verificado suba la codificación a su Arduino
- Desconecte el Arduino y realice la siguiente conexión

PROCEDIMIENTO:

1. Conectamos dos LED a nuestra Protoboard como se muestra la imagen ánodo con la resistencia y cátodo a tierra.
2. Conectamos tres resistencias de 220 ohm a cada LED.
3. Conectamos la resistencia al pin 8 el LED rojo y al pin 9 al LED amarillo y el pin 10 al LED amarillo.
4. Verifique q está bien conectado todo espere la autorización del docente y conecte su Arduino a la fuente

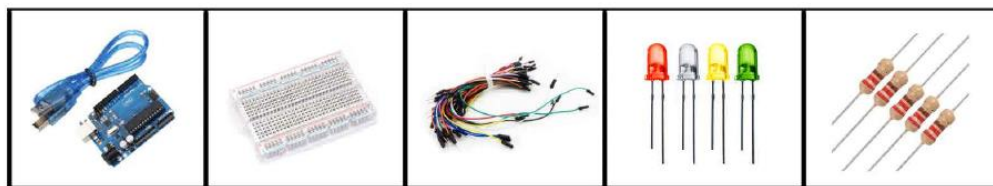


Una vez que estamos seguro que ah sido conectado de forma correcta procedemos con el funcionamiento y análisis del circuito.

4.5. Control de 1 LED con Arduino opción 2

OBJETIVO: controlar 1 LED emitiendo una señal HIGH (1) o LOW (0) a través de 1 pin digital del Arduino utilizando int.

MATERIALES:



Placa Uno / Placa de prueba / Cable de puente / 1 LED / 1 Resistencia de 220 ohmios (Ω), cables para conexión.

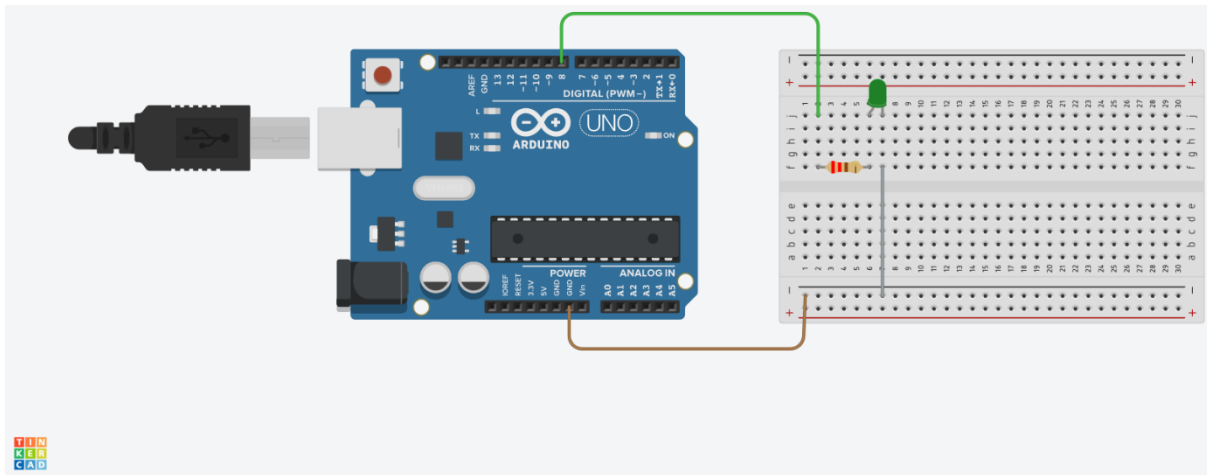
CODIFICACIÓN:

```
int verde=8;

void setup()
{
  pinMode(verde, OUTPUT);
}

void loop()
{
  digitalWrite(verde,HIGH);
  delay (3000);
  digitalWrite(verde,LOW);
  delay (3000);
}
```

Una vez que estés seguro que el circuito ha sido cargado con éxito en la placa Arduino uno procedemos con el armado del circuito que esta en la siguiente imagen



4.6. Control de 2 LED con Arduino opción 2

OBJETIVO: controlar 2 LED emitiendo una señal HIGH (1) o LOW (0) a través de 2 pines digitales del Arduino utilizando int.

MATERIALES:



Placa Uno / Placa de prueba / Cable de puente / 2 LED / 2 Resistencia de 220 ohmios (Ω), cables para conexión.

CODIFICACIÓN:

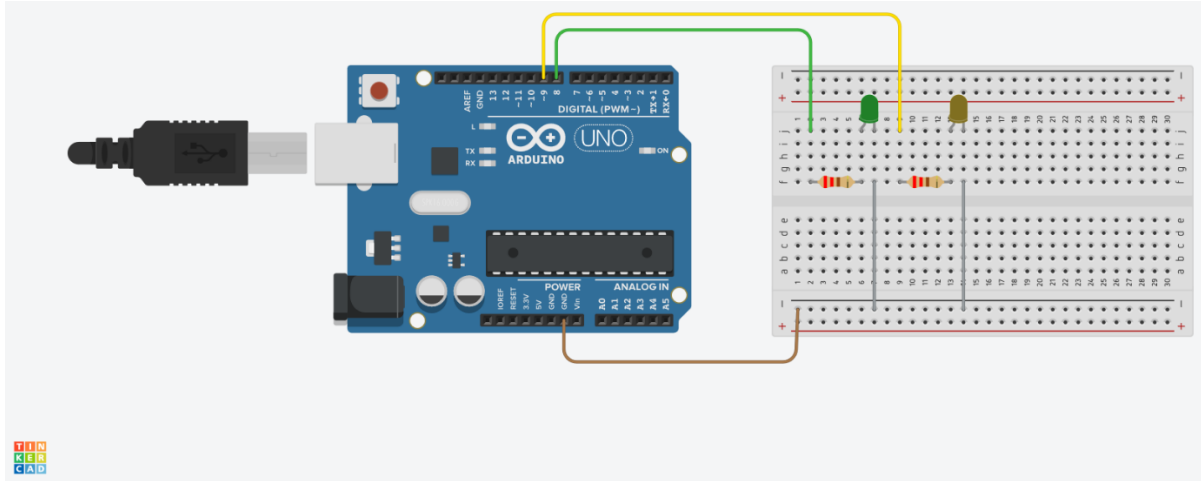
```
int verde=8;
int amarillo=9;

void setup()
{
  pinMode(verde, OUTPUT);
  pinMode(amarillo, OUTPUT);
}

void loop()
{
  digitalWrite(verde,HIGH);
  digitalWrite(amarillo,LOW);
  delay (6000);

  digitalWrite(amarillo,HIGH);
  digitalWrite(verde,LOW);
  delay (2000);
}
```

Una vez que estés seguro que el circuito ha sido cargado con éxito en la placa Arduino uno procedemos con el armado del circuito que esta en la siguiente imagen



4.7. Control de 3 LED con Arduino opción 2

OBJETIVO: controlar 3 LED emitiendo una señal HIGH (1) o LOW (0) a través de 3 pines digitales del Arduino utilizando int.

MATERIALES:



Placa Uno / Placa de prueba / Cable de puente / 3 LED / 2 Resistencia de 220 ohmios (Ω), cables para conexión.

CODIFICACIÓN:

```
int verde=8;
int amarillo=9;
int rojo=10;

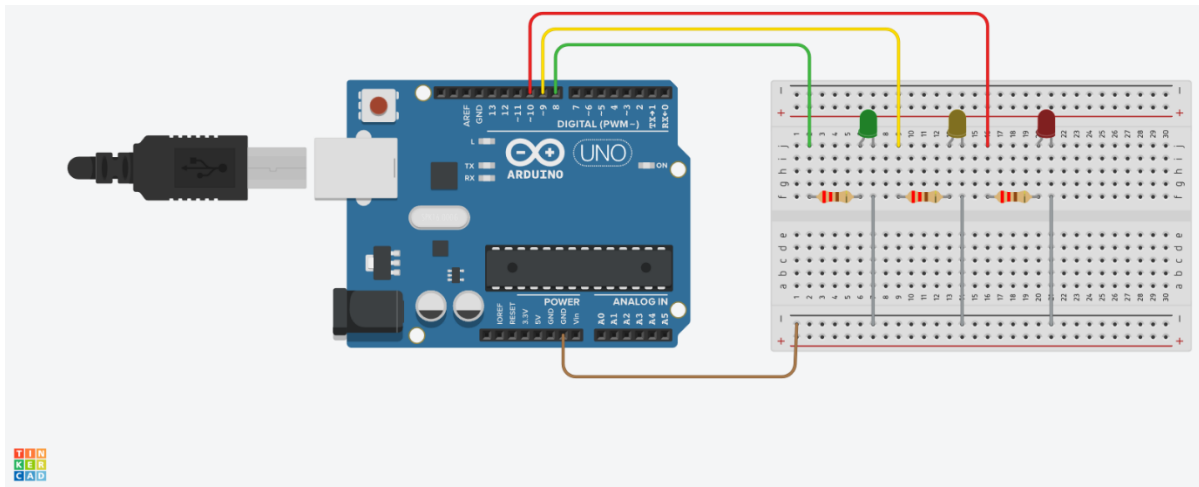
void setup()
{
    pinMode(verde, OUTPUT);
    pinMode(amarillo, OUTPUT);
    pinMode(rojo, OUTPUT);
}

void loop()
{
    digitalWrite(verde,HIGH);
    digitalWrite(amarillo,LOW);
    digitalWrite(rojo,LOW);
    delay (6000);

    digitalWrite(amarillo,HIGH);
    digitalWrite(verde,LOW);
    digitalWrite(rojo,LOW);
    delay (2000);
}
```

```
digitalWrite(rojo,HIGH);
digitalWrite(amarillo,LOW);
digitalWrite(verde,LOW);
delay (4000);
}
```

Una vez que estés seguro que el circuito ha sido cargado con éxito en la placa Arduino uno procedemos con el armado del circuito que esta en la siguiente imagen



Lo que aprendimos

- ✓ Se aprendió sobre la placa Arduino y el sw de Tinkercad y Arduino.
- ✓ Conocimos los dispositivos de entrada y salida Arduino, codificamos las funciones de entrada y salida conociendo cada una de ellas por su nombre, apariencia física y tipo.
- ✓ Elaboramos los ejercicios planteados de manera correcta,
- ✓ elaboramos circuitos en Tinkercad y codificación en Arduino, aplicando todos los conocimientos facilitados por el profesor.

Prueba evaluativa de conocimiento

Instrucciones: A continuación, se formulan una serie de preguntas, lea detenidamente cada una de ellas y seleccione la opción que usted considera correcta.

1. ¿Es una variación de voltaje entre $-V_{cc}$ a $+V_{cc}$? sin pasar por los valores intermedios?
 - a) Señal digital
 - b) Señal análoga
 - c) Arduino
2. ¿Es una variación de voltaje entre $-V_{cc}$ a $+V_{cc}$? sin pasar por los valores intermedios?
 - a) Señal digital
 - b) Señal análoga
 - c) Arduino
3. ¿Es un dispositivo de salida Arduino?
 - a) Pulsador
 - b) Potenciómetro
 - c) Motor



4. ¿Al colocar `pinMode(8, OUTPUT)` estamos declarando que?
 - a) El pin como entrada
 - b) El pin como salida
 - c) El pin con voltaje 0V
5. ¿Al colocar `digitalWrite(8, LOW)` estamos declarando que?
 - a) Tendremos 0V en el pin
 - b) El pin como entrada
 - c) El pin con 5V

Respuesta breve:

1. ¿Nombre 10 dispositivos de salida?
2. ¿Nombre 10 dispositivos de entrada?
3. ¿Cuáles son las funciones más utilizadas para codificar entradas y salidas en Arduino?
4. ¿Enumere los pasos más importantes para poder subir un código Arduino?

Tipo practico:

Cada circuito deberá ser trabajado en Tinkercad

Materiales:

1. Computadora
2. Programa Arduino
3. Tinkercad
1. ¿Realice el código de programación y diseño de circuito para Arduino, del encendido de 4 led con Arduino?
2. ¿Realice el código de programación y diseño de circuito para Arduino, del encendido de 5 led con Arduino?
3. ¿Realice el código de programación y diseño de circuito para Arduino, del encendido de 6 led con Arduino?



HONDURAS

GOBIERNO DE LA REPÚBLICA



Educación

Gobierno de la República



MT: Kira Sánchez Sánchez



*MT: Jimmy Reniery Velásquez
Velásquez*



MT: Alejandro Ismael Rojas